Heurísticas gulosas para o SCP

Humberto J. Longo

11 de julho de 2018

1 Recobrimento de um Conjunto

O problema de recobrimento de um conjunto é equivalente à busca pelo menor número de subconjuntos, de um determinado conjunto, que unidos geram esse conjunto principal. De maneira precisa tem-se que, dado um conjunto $I = \{1, \ldots, m\}$ e uma família $F = \{I_1, \ldots, I_n\}$ de subconjuntos de I associada a $J = \{1, \ldots, n\}$, qualquer subconjunto $J^* \subseteq J$ define um recobrimento de I, se $\bigcup_{j \in J^*} I_j = I$ [8, 3]. Um recobrimento J^* é dito redundante se $J^* - \{j\}$ $(j \in J^*)$ ainda define um recobrimento. Um caso especial de recobrimento, dito particionamento de I, é obtido quando J^* satisfaz $I_j \bigcap I_k = \emptyset$ para $j, k \in J^*$ $(j \neq k)$.

Associando-se a cada conjunto j da família F um custo c_j, um recobrimento J^* terá um custo total de $\sum_{j \in J^*} c_j$. O interesse é encontrar um recobrimento de custo mínimo. Este problema de minimização é chamado de Problema de Recobrimento de um Conjunto (*SCP - Set Covering Problem*) [14]. Os custos c_j são sempre positivos e em geral distintos. Entretanto, uma classe de instâncias muito estudada é aquela em que todos os custos c_j são iguais a 1 (*unicost*). Neste texto será tratado o SCP geral, mas devido a importância do SCP de custo unitário, eventualmente serão feitas referências específicas a esse caso especial.

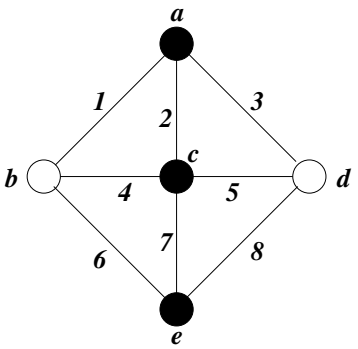
O Problema de Recobrimento de um Conjunto é computacionalmente difícil de ser resolvido e claramente pertencente à classe dos problemas NP-difíceis. Karp [13] mostrou que o problema de se encontrar uma cobertura dos vértices de um grafo G reduz-se trivialmente ao problema de se encontrar um recobrimento de um conjunto I, do seguinte modo: dado um grafo $G = (V, E)$, para $u, v \in V$, o elemento $(u, v) \in I$ se existir o arco $(u, v) \in E$ e $I_j \subseteq I$ for o conjunto de arcos incidentes ao vértice j.

A relação entre o Problema de Recobrimento de um Conjunto e o de Cobertura dos Vértices de um Grafo pode ser mais facilmente compreendida observando-se o exemplo a seguir. A Figura 1.(a) mostra uma cobertura de vértices de um grafo e a Figura 1.(b) mostra um equivalente recobrimento de conjunto.

Reforçando a dificuldade da resolução do Problema de Recobrimento de um Conjunto, Garey e Johnson [9] observam que, mesmo se $|I_j| \leq 3,\ \forall\ I_j \subseteq I$, o problema continua pertencendo à classe $\mathcal{NP}$-Completo. Só é possível garantir que seja resolvido em tempo polinomial se $|I_j| \leq 2,\ \forall\ I_j \subseteq I$. Os métodos propostos por Norman e Rabin [17] e Edmonds [4], por exemplo, resolvem este caso especial de forma bastante eficiente.

2 Modelo IP do SCP

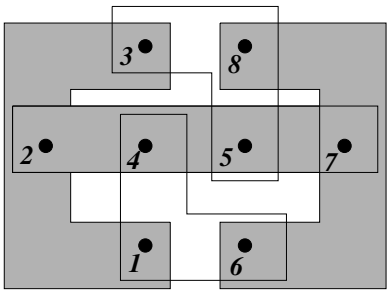
O Problema de Recobrimento de um Conjunto pertence à subclasse dos problemas de Programação Linear Inteira 0-1. Para a formalização do SCP as variáveis $x_j, a_{ij}\ e\ b_i$, para $j = 1, \ldots, n$ e $i = 1, \ldots, m$ são definidas como:



G = ({a,b,c,d,e},{1,2,3,4,5,6,7,8})

(a)

I = {1,2,3,4,5,6,7,8}
Ia = {1,2,3}
Ib = {1,4,6}
Ic = {2,4,5,7}
Id = {3,5,8}
Ie = {6,7,8}



(b)

Figura 1: Exemplos de cobertura dos vértices de um grafo e de recobrimento de um conjunto.

$$x_j = \begin{cases} 1 & \textit{se o conjunto } j \textit{ está no recobrimento}, \\ 0 & \textit{caso contrário}; \end{cases}$$

$$a_{ij} = \begin{cases} 1 & \textit{se o elemento } i \textit{ pertence ao conjunto } j, \\ 0 & \textit{caso contrário}; \end{cases}$$

$$b_i = 1.$$

O Problema de Recobrimento de um Conjunto pode agora ser escrito como segue:

$$\begin{array}{lrcll} \min \sum_{j=1}^{n} c_j x_j & & & & \\ \text{s.a.} \sum_{j=1}^{n} a_{ij} x_j & \geq & 1 & i = 1, \ldots, m \\ x_j & \in & \{0,1\} & j = 1, \ldots, n \end{array} \tag{1}$$

Em algumas partes do texto, será mais conveniente discutir o SCP em termos de conjuntos e recobrimentos do que desigualdades e variáveis. As definições a seguir fornecem o suporte teórico básico para a discussão do SCP em termos de conjuntos:

- $I = \{1, \ldots, m\}$: índices referentes às restrições;
- $J = \{1, \ldots, n\}$: índices referentes às variáveis;
- $I_j = \{i \in I \mid a_{ij} = 1\},\ j \in J$: restrições nas quais a variável x_j aparece; e
- $J_i = \{j \in J \mid a_{ij} = 1\},\ i \in I$: variáveis contidas na restrição i.

2.1 SPP e SP

Considerando-se o SCP como o modelo abstrato descrito na Seção 2, existem dois outros problemas diretamente ligados a ele: o Problema de Particionamento de um Conjunto (*SPP - Set Partitioning Problem*) e o (*SP - Set Packing Problem*).

Usando-se a notação definida na Seção 2, o SP reduz-se a se encontrar o maior conjunto J^* tal que $\bigcup_{j \in J^*} I_j \subseteq I$ e $\bigcap_{j \in J^*} I_j = \emptyset$, ou seja, o problema é unir o máximo de subconjuntos da família F sem que haja interseção entre eles. O SPP é um caso especial do SCP que é obtido quando um recobrimento J^* satisfaz $I_j \bigcap I_k = \emptyset$ para $j, k \in J^*$ $(j \neq k)$,

ou seja, o problema é encontrar um recobrimento onde a interseção entre os subconjuntos seja vazia.

Analogamente ao SCP, esses dois últimos podem ser modelados como problemas de programação linear inteira. Observando-se que no SP o objetivo é maximizar o número de subconjuntos utilizados, sua formalização é a seguinte:

$$\begin{aligned} \max & \sum_{j=1}^{n} c_j x_j \\ \text{s.a.} & \sum_{j=1}^{n} a_{ij} x_j \leq 1 \qquad i = 1, \ldots, m \\ & x_j \in \{0,1\} \quad j = 1, \ldots, n \end{aligned} \tag{2}$$

e a formalização do SPP:

$$\begin{aligned} \min & \sum_{j=1}^{n} c_j x_j \\ \text{s.a.} & \sum_{j=1}^{n} a_{ij} x_j = 1 \qquad i = 1, \ldots, m \\ & x_j \in \{0,1\} \quad j = 1, \ldots, n \end{aligned} \tag{3}$$

As formalizações do SCP e do SP como problemas de programação linear inteira são bastante semelhantes e contêm uma importante propriedade. Sendo os dois problemas de custos unitários, então relaxando-se a condição binária nas variáveis em ambos, tem-se que a relaxação do SP é o dual da relaxação do SCP.

Certas equivalências entre os três problemas são observadas com a adoção do modelo matemático descrito. O SPP pode ser convertido tanto para o SCP quanto para o SP. A conversão do SPP para SCP é descrita na próxima Seção (2.2) e a conversão do SPP para o SP é análoga.

2.2 Conversão de SPP a SCP

Todo SPP com solução factível pode ser convertido em um SCP equivalente. No contexto, equivalência significa ambos terem a mesma solução ótima. O procedimento de conversão e sua demonstração do mesmo foram extraídos de Lemke, Salkim e Spielberg [15], Garfinkel e Nemhauser [10] e Taha [20].

O procedimento de conversão dos problemas consiste na simples troca do vetor de custos c e, obviamente, na mudança das equações para inequações. Para tanto, sejam $t_j = \sum_{i=1}^{m} a_{ij},\ j = 1, \ldots, n$, e L um número tal que $L > \sum_{j=1}^{n} c_j$. O SCP obtido de um SPP com a mudança do vetor de custos c para $c'_j = c_j + Lt_j, j = 1, \ldots, n$, possui o mesmo conjunto de soluções ótimas do problema original. A demonstração desse procedimento mostra que qualquer solução factível para o SCP, e infactível para o SPP, tem valor maior do que a melhor solução factível para o SPP.

Sejam S_c e S_p os conjuntos de soluções para o SCP e SPP, respectivamente, observe que $S_c \supset S_p$ (se $S_c = S_p$ a demonstração é trivial). Tomando-se quaisquer $x^p \in S_p$ e $x^c \in S_c - S_p$, então:

$$\sum_{j=1}^{n} c'_j x_j^c = \sum_{j=1}^{n} c_j x_j^c + L \sum_{j=1}^{n} t_j x_j^c$$

Por definição:

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} x_j^c &\geq 1 \qquad i = 1, \ldots, m \\ &= 1 + \Delta_i \end{aligned}$$

Logo:

$$\sum_{j=1}^{n} c'_j x_j^c \geq L \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j^c) = L \sum_{i=1}^{m} (1 + \Delta_i) \geq L(m+1)$$

Por outro lado:

$$\begin{aligned} \sum_{j=1}^{n} c'_j x_j^p &= \sum_{j=1}^{n} c_j x_j^p + L \sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij} x_j^p) \\ &= \sum_{j=1}^{n} c_j x_j^p + Lm \leq \sum_{j=1}^{n} c_j + Lm < L + Lm = L(m+1) \end{aligned}$$

Isto mostra que $c'x^c > c'x^p$, significando que uma solução factível apenas para o SCP não pode ser ótima para o SPP.

Esse procedimento de conversão admite que o SPP tem solução factível, o que nem sempre é correto, ao contrário do SCP. Como pode ser difícil determinar de antemão se um SPP é infactível, é necessário verificar, após a resolução do SCP equivalente, se a solução obtida é factível para o SPP.

O procedimento anterior permite a resolução de instâncias do SPP por meio de um algoritmo especialmente construído para o SCP, contudo, pode acarretar em sérios erros de arredondamento na implementação computacional. Esse problema é particularmente suscetível de ocorrer na soma de custos grandes, relativamente aos custos individuais, ou seja, quando o vetor de custos é composto por valores pequenos e próximos uns dos outros [19].

3 Resolução Heurística do SCP

Existe na literatura um grande número de heurísticas de construção para o SCP. Essas heurísticas são essencialmente gulosas. O procedimento geral seleciona repetidamente, com base em algum critério, uma variável j que receberá valor 1. A parada se dá quando uma solução factível é obtida.

Considerando-se que o critério a ser usado para a escolha de uma variável é uma certa função f do coeficiente de x_j e da cardinalidade do conjunto I_j de linhas recobertas pela variável x_j, tais heurísticas têm a seguinte forma geral:

- **Heurística gulosa primal**

  1. $N = \{1, \ldots, n\}$;
     $x_j = 0$ e $I_j(x) = I_j,\ j \in N$;
  2. Se $I_j(x) = \emptyset$ para todo $j \in N$ então PARE (o vetor x é uma solução factível); caso contrário, encontre $k \in N$ que maximize a função $f(c_k, I_k(x))$;
  3. Faça $x_k = 1,\ N = N - \{k\}$ e $I_j(x) = I_j(x) - I_k(x),\ j \in N$;
     Retorne ao passo 2.

O uso de uma função diferente para a escolha de uma variável corresponde a uma heurística diferente. Chvátal [2] propôs a função $f(c_j, I_j(x)) = c_j/|I_j(x)|$, escolhendo, em outras palavras, a variável que recobre o maior número de restrições, dentre as não recobertas, por unidade de custo. No caso em que todos os custos são unitários, ou seja, $c_j = 1,\ \forall\ j \in J$, a heurística proposta por Chvátal reduz-se às propostas por Johnson [12] e Lovász [16].

Balas e Ho [1] analisaram essa mesma função e quatro outras ($c_j, c_j/\log_2|I_j(x)|$, $c_j/(|I_j(x)|\log_2|I_j(x)|)$ e $c_j/(|I_j(x)|\ln|I_j(x)|)$). Vasko e Wilson [21] proposeram duas outras funções ($c_j/(|I_j(x)|)^2$ e $\sqrt{c_j}/(|I_j(x)|)^2$). Uma variação, também proposta por Vasko

e Wilson, é o uso intercalado de várias funções de escolha das variáveis na construção de uma mesma solução. Na heurística proposta por Roth [18], a função de escolha das variáveis foi eliminada, sendo a escolha feita de modo aleatório.

Desse conjunto de heurísticas citado anteriormente, uma possível subclasse, é aquela formada por heurísticas que limitam as variáveis passíveis de serem escolhidas em cada iteração a um subconjunto do total de variáveis. O uso de diferentes critérios para a escolha de tais subconjuntos gera essa subclasse de heurísticas.

Balas e Ho [1] usaram uma heurística dessa subclasse onde cada subconjunto era formado pelas variáveis que compunham cada restrição do problema. O critério de escolha de um subconjunto foi o número de elementos em cada um, sendo escolhido sempre o de menor cardinalidade. Outra modalidade de escolha de subconjuntos de variáveis foi proposta por Feo e Resende [5], onde a cada passo de escolha de uma variável, esta era escolhida dentre aquelas que recobriam um número mínimo de restrições.

Alguma inovação foi introduzida nessa classe de heurísticas por Fisher e Kedia [7], com a utilização de uma solução do problema dual da relaxação contínua do SCP para determinar o impacto das restrições do problema na escolha de variáveis. A Heurística proposta por eles segue o mesmo padrão da proposta por Balas e Ho, ou seja, em cada iteração uma variável é escolhida de um subconjunto (variáveis em uma restrição) do conjunto de variáveis. O subconjunto a ser escolhido é aquele que maximiza $u_i|J_i|$, onde u_i é a variável dual associada a restrição i e J_i é o conjunto de variáveis da restrição i. A regra para escolha de uma variável dentro de um subconjunto é similar à usada por Chvátal [2], Lovász [16] e Johnson [12], exceto que c_j é trocado por $c_j - \sum_{i \in I_j} u_i$, onde I_j é o conjunto de restrições recobertas pela variável j.

A análise de desempenho de uma heurística para o SCP, pertencente à classe geral citada anteriormente, foi feita inicialmente por Chvátal [2]. Chvátal mostrou que $Z_{HEU}/Z^* \leq \sum_{j=1}^{d} 1/j$ é o melhor limite possível, onde $d = \max_{j \in J} |I_j|$ e Z^* e Z_{HEU} são os valores da solução ótima e da encontrada pela heurística, respectivamente. Posteriormente Ho [11] mostrou que o desempenho, no pior caso, de qualquer heurística dessa classe geral é dominado por aquele mostrado por Chvátal [2]. Outras análises nesta área podem ser encontradas em [1, 6, 9, 12, 16, 22].

Bibliografia

[1] E. Balas and A. Ho. Set Covering Algorithms using Cutting Planes, Heuristics, and Subgradient Optimization: A Computational Study. *Mathematical Programming Study*, 12:37–60, April 1980.

[2] V. Chvátal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235, August 1979.

[3] J. Edmonds. Covers and Packing in a Family of Sets. *Bulletin of American Mathematical Society*, 68(5):494–499, September 1962.

[4] J. Edmonds. Path, Trees, and Flowers. *Canadian Journal of Mathematics*, 17(3):449–467, 1965.

[5] T. A. Feo and M. G. C. Resende. A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem. *Operations Research Letters*, 8:67–71, April 1989.

[6] M. L. Fisher. Worst-Case Analysis of Heuristic Algorithms. *Management Science*, 26(1):1–17, January 1980.

[7] M. L. Fisher and P. Kedia. Optimal Solution of Set Covering/Partitioning Problems using Dual Heuristics. *Management Science*, 36(6):674–688, June 1990.

[8] D. R. Fulkerson and H. J. Ryser. Widths and Heights of (0,1)-Matrices. *Canadian Journal of Mathematics*, 13(2):239–255, 1961.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[10] R. S. Garfinkel and G. L Nemhauser. *Integer Programming*. John Wiley & Sons, New York, 1972.

[11] A. C. Ho. Worst Case Analysis of a Class of Set Covering Heuristics. *Mathematical Programming*, 23(2):170–180, June 1982.

[12] D. S. Johnson. Approximation Algorithms for Combinatorial Problems. *Journal of Computer and System Sciences*, 9(3):256–278, December 1974.

[13] R. M. Karp. Reducibility Among Combinatorial Problems. In *Complexity of Computer Computations*, pages 85–104, New York, 1972. Plenum Press.

[14] E. L. Lawler. Covering Problems : Duality Relations and a New Method of Solution. *SIAM Journal Applied Mathematics*, 14(5), September 1966.

[15] C. Lemke, H. Salkin, and K. Spielberg. Set Covering by Single Branch Enumeration. *Operations Research*, 19(4):998–1022, July 1971.

[16] L. Lovász. On the Ratio of Optimal and Fractional Covers. *Discrete Mathematics*, 13(4):383–390, December 1975.

[17] R. Z. Norman and M. O. Rabin. An Algorithm for a Minimum Cover of a Graph. *Proceeding of the American Mathematical Society*, 10(2):315–319, 1959.

[18] R. Roth. Computer Solutions to Minimum-Cover Problems. *Operations Research*, 17(3):455–465, May 1969.

[19] H. M. Salkin and K. Mathur. *Foundations of Integer Programming*. North-Holland, 1989.

[20] H. A. Taha. *Integer Programming: Theory, Applications and Computations*. Academic Press, 1975.

[21] F. J. Vasko and G. R. Wilson. An Efficient Heuristic for Large Set Covering Problems. *Naval Research Logistics Quaterly*, 31(2):163–171, May 1984.

[22] L. A. Wolsey. Heuristic Analysis, Linear Programming and Branch and Bound. *Mathematical Programming Study*, 13:121–134, August 1980.